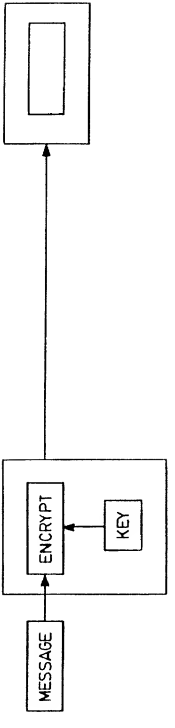



FIG. 1

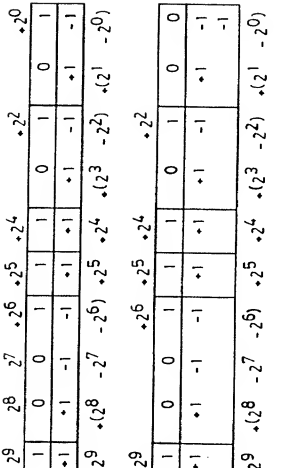
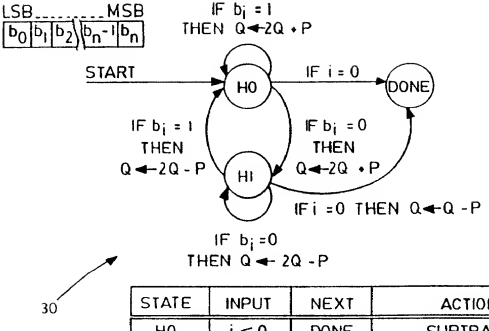

629 = $2^9$ $2^8$ $2^7$ $+2^6$ $+2^5$ $+2^4$ $+2^2$ $+2^0$
1 0 0 1 1 0 1
+1 +1 -1 -1 +1 +1 -1 +1 -1
$2^9 + (2^8 - 2^7 - 2^6) + 2^5 + 2^4 + (2^3 - 2^2) + (2^1 - 2^0)$
628 = $2^9$ $+2^6$ $+2^5$ $+2^4$ $+2^2$
1 0 0 1 1 0 1 0 0
+1 +1 -1 -1 +1 +1 -1 +1 -1 -1
$2^9 + (2^8 - 2^7 - 2^6) + 2^5 + 2^4 + (2^3 - 2^2) + (2^1 - 2^0)$

FIG. 2




| STATE | INPUT | NEXT | ACTION |
|---|---|---|---|
| H0 | $i < 0$ | DONE | SUBTRACT |
| H0 | $b_i = 0$ | H1 | DOUBLE, ADD |
| H0 | $b_i = 1$ | H0 | DOUBLE, ADD |
| H1 | $i < 1$ | DONE | |
| H1 | $b_i = 0$ | H1 | DOUBLE, SUBTRACT |
| H1 | $b_i = 1$ | H0 | DOUBLE, SUBTRACT |

FIG. 3

```
BEGIN :
    i  := N          ; START FROM MSB                    L1
    Q  := 0          ; INITIALIZE ACCUMULATOR            L2
    H  := 0          ; INITIALIZE STATE                  L3

LOOP :               ; FOR ALL BITS
    Q  := Q + Q      ; DOUBLE ACCUMULATOR                L4

    IF H = 0         ; IF H STATE IS SET                 L5
      Q  := Q + P    ;   ADD BASE POINT TO ACCUMULATOR   L6
      GOTO ENDLOOP   ;                                   L7
    ELSE             ; ELSE
      Q  := Q + (-P) ;   SUBTRACT BASE POINT             L8
      GOTO ENDLOOP   ;                                   L9

ENDLOOP :
    H  := b[i]       ; SET H STATE TO VALUE OF b[i]      L10
    i  := i -1       ; PROCESS NEXT BIT                  L11
    IF i>0           ; IF BIT EXISTS                     L12
      GOTO LOOP      ;   CONTINUE AT TOP OF LOOP        L13

    IF H = 0         ; IF EXISTING FROM H = 0 STATE      L14
      Q  := Q + (-P) ;   CORRECT RESULT BY FINAL SUBTRACT L15
    END                                                  L16
```

FIG. 4

```
BEGIN :
        i : = N          ; START FROM MSB                    LL1
        Q : = 0          ; INITIALIZE ACCUMULATOR            LL2

H0 :                     ; STATE ENTRY POINT
        Q : = Q + Q      ; DOUBLE ACCUMULATOR                LL3
        Q : = Q + P      ; ADD BASE POINT TO ACCUMULATOR     LL4
        GOTO ENDLOOP     ; BRANCH TO END OF LOOP TESTS       LL5

H1 :                     ; STATE ENTRY POINT
        Q : = Q + Q      ; DOUBLE ACCUMULATOR                LL6
        Q : = Q + (-P)   ; SUBTRACT BASE POINT FROM
                         ; ACCUMULATOR                       LL7
        GOTO ENDLOOP     ; BRANCH TO END OF LOOP TESTS       LL8

ENDLOOP :                ; END OF LOOP TESTS
        IF b[i]=1        ; IF CURRENT BIT IS SET             LL9
          GOTO NEXT H0   ;   FOLLOW H0 PATH                  LL10
                         ; ELSE FALL INTO H1 PATH

NEXT H1 :                ; H1 PATH
        i : =i-1         ; PROCESS NEXT BIT                  LL11
        IF i>0           ; IF BIT EXISTS                     LL12
          GOTO H1        ;   EXECUTE H1 STATE                LL13
        Q : = Q + (-P)   ; ELSE CORRECT RESULT AND END       LL14
        END                                                  LL15

NEXT H0 :                ; H0 PATH
        i : =i-1         ; PROCESS NEXT BIT                  LL16
        IF i>0           ; IF BIT EXISTS                     LL17
          GOTO H0        ;   EXECUTE H0 STATE                LL18
        END              ; ELSE END                          LL19
```

FIG. 5

```
BEGIN :
        i : = N
        Q : = 1

H0 :
        Q : = Q·Q  (Q^2)
        Q : = Q·M
        GOTO ENDLOOP

H1 :
        Q : = Q·Q
        Q : = Q/M  (Q-M-1)

ENDLOOP :
        IF b[i] = 1   GOTO ENDLOOP

NEXT H1 :
        i = i - 1
        IF i>0
        GOTO H1
        Q = Q/M
        END

NEXT H0 :
        i = i - 1
        IF i>0
        GOTO H0
        END
```

60

FIG. 6

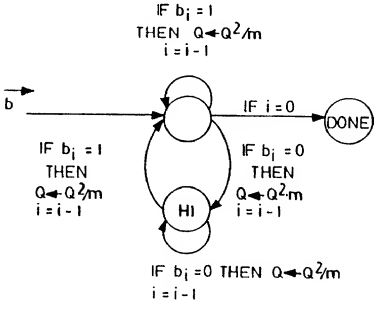

IF $b_i = 1$
THEN $Q \leftarrow Q^2/m$
$i = i - 1$
b
IF $i = 0$
DONE
IF $b_i = 1$
THEN
$Q \leftarrow Q^2/m$
$i = i - 1$
IF $b_i = 0$
THEN
$Q \leftarrow Q^2 m$
$i = i - 1$
HI
IF $b_i = 0$ THEN $Q \leftarrow Q^2/m$
$i = i - 1$

FIG. 7

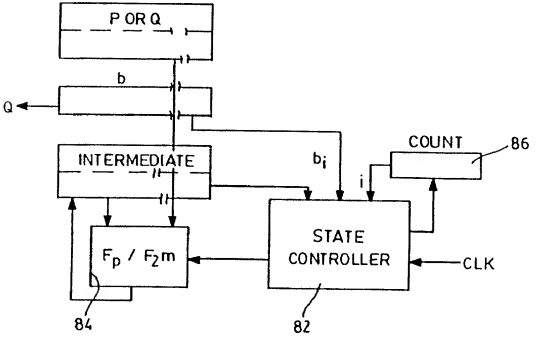

P OR Q
b
Q
INTERMEDIATE
$b_i$
COUNT
86
i
STATE
CONTROLLER
CLK
$F_p$ / $F_2m$
84
82

FIG. 8